

Fenceless obstacle avoidance method for efficient and safe human–robot collaboration in a shared work space

Mohammad Mayyas , Sai P Vadlamudi and Muhammed A Syed

Abstract

In a given manufacturing setting where workers or robots are coexisting in a confined area and their movements are not coordinated due to loss in communication or because they are freely ranging relative to each other, the development of an onboard safeguard system for a robot becomes a necessity to reduce accidents while the production efficiency is uncompromised. This article develops a two-dimensional dynamics model that predicts the relative position between a robot's end-of-arm tooling and an approaching object or threat. The safety strategy applied to the robot is derived from the calculation of three parameters: the time of collision predicted from the linear motion between the approaching object and the robot's end-of-arm tooling, the relative absolute distance, and the overlapping area ratio. These parameters combined are updated in a cost function that is sufficiently alarming the collision severity of an approaching object in real time. This model enables deployment a safe and a productive collaborative interaction in the manufacturing environment where workers and robots are seemingly moving in close proximity within an open workspace with less safeguard barriers.

Keywords

Collaborative robotics, HRI, obstacle avoidance, worker safety, manufacturing work space

Date received: 03 June 2020; accepted: 16 August 2020

Topic Area: AI in Robotics; Human Robot/Machine Interaction

Topic Editor: Andrey V Savkin

Associate Editor: Istvan Harmati

Introduction

There is a need to develop predictive collision methods to improve worker–robot safety due to the increase installation of industrial and service robotics system in factories worldwide, where 1.4 million new industrial robots were expected by 2019 attributable to the use of autonomous guided systems.¹ Over the years, assessment methods have been deployed to improve the safety and reliability of robots serving several industry sectors including production, warehouses and logistics, autonomous transportation, and health-care assistive mobile robots.² Occupational accidents have been reporting worker injuries—those due to engineering and human errors³ such as lack or absence of safeguard system to stop or slow down the operation of

robot when a worker accidentally enters robots work envelop, modifications to machinery program, or bypassing safeguards as well as inexperience of workers. The occupational machine-related accidents accounted for over 5000 fatalities in the United States during 2003–2010, of which 23% of fatalities were due to stationary machinery and 77% caused by mobile machinery. Manufacturing and

Mechatronics Engineering Technology, Bowling Green State University, Bowling Green, OH, USA

Corresponding author:

Mohammad Mayyas, Mechatronics Engineering Technology, Bowling Green State University, Bowling Green, OH.43402, USA.

Email: mmayyas@bgsu.edu



Creative Commons CC BY: This article is distributed under the terms of the Creative Commons Attribution 4.0 License (<https://creativecommons.org/licenses/by/4.0/>) which permits any use, reproduction and distribution of the work without further permission provided the original work is attributed as specified on the SAGE and Open Access pages (<https://us.sagepub.com/en-us/nam/open-access-at-sage>).

construction were among top hazard industries accounting for 14% and 22% of the fatalities, respectively.⁴ In the United States alone, the Bureau of Labor Statistics revealed 717 fatal work injuries occurred during the year 2013 as a result of direct contact with object.⁵ Reports in Canada and in the Netherlands revealed that most accidents are caused by operators entering hazardous zone of machinery during maintenance and the handling of production disturbances.^{6,7} The worker safety is typically monitored by speed and separation from a robot which together used to detect collision in advance by particular reaction strategies and safeguards tools.⁸ The tools include non-contact ultrasonic position sensors, contact capacitive or piezoelectric force and pressure sensors, door interlock switch, virtual barriers enabled by laser curtain, restricted work envelope that allows program robot movement within the desired workspace, pressure mat to detect the presence of operator, optical cameras, motion capture systems, and many others. Moreover, one common fenceless technology is laser scanners which have been traditionally used to safeguard a predefined number of zones around dangerous machinery.⁹ Position-based collision mitigation are applied today in cyber-physical system where robot workspace is typically predefined into three safety zones to detect the human presence or moving machinery like AGV's. These technologies focus on safety first regardless of the efficiency of the manufacturing process. For example, wearable surface pressure sensor array on collaborative robot, such as MRK-Systeme KR SI, Fanuc CR-35iA, ABB YuMi, UR5, and KUKA LBR iiwa may lead to a reduced robot speed or entire motion stop at object detection.¹⁰ The common practice is to enable all safety tools regardless of its effect on the production rate. Of course, while priority number one is to protect people in workplace, it is not yet clearly understood the trade-offs between safety and efficiency of robot.

Related work

The production efficiency in advanced manufacturing environment might be improved through the increase of the number of collaborative robot installation, however, that comes at the cost of risking the safety of coexistence with the workers. Particularly, "just-in-time sensing" based safety methods might not be adequate to address the issue of downtime in production when accidents or interference have already occurred. On the other hand, predictive collision-free algorithms—such as dynamic programming, genetic algorithm, and distance transform methods¹¹ are commonly used to optimize the path and the speed of mobile robots in the presence of obstacles. Similarly, to improve the efficiency of the manufacturing and the safety between human and industrial robot arm in a shared workplace, the path of a worker roaming around end-of-arm tooling (EOAT) of robot can be predicted at an instance

and kept updated in real time with goal to estimate the future level of collision before accidents happen.

The goal of robot path planning in dynamic environment is to create a collision-free route more accurately and timely. Masehian et al. survived over 1381 articles covering classic and heuristic approaches in robot motion planning within time span 1973 to 2007.¹² Classic methods were popular prior to 1982 and it uses general approaches such as roadmap, cell decomposition, potential fields, and mathematical programming. Classical methods such as PF, DWA, A*, and PRM path planning are easy to implement, perform better when blended with other methods, fast with respect to response time.^{13,14} However, such conventional methods are computationally expensive and typically rely on accuracy of cameras and sensors to collect data from the environment; making measurement uncertain when noise are inherent.¹⁵ For example, a heuristic A-Star and dynamic steering algorithms use predefined grid-based environment map to find the shortest path based on information received from sonar sensor.¹⁶ Distance remote control algorithm was developed to predict and avoid collisions between vehicles by maintaining a given safe distance between the robot and the obstacle.¹⁷ Potential fields were first introduced by Oussama Khatib in 1986, where an attractive potential pulls the robot toward the goal configuration, and a repulsive potential would push the robot away from the obstacles.¹⁸ The mathematical programming approach formulates an optimization from a set of inequalities on the configuration parameters that minimize collision. The dimension of the configuration space implemented in cell-decomposition is large and time-consuming.¹⁹

Nikolakis et al. developed system that used depth data and tool center point coordinate system of a robot to calculate the distance between the robot's end-effectors and worker in the workspace to control the speed of the robot and send notifications or alerts.²⁰ Long et al. developed a three-dimensional (3D) dynamic security zone for a collaborative robot UR10 to increase its performance in terms of average velocity by using three post-collision safety laser system.²¹ Object detection around this robot was divided into three security modes: nominal mode, reduced mode, and passive mode, which change automatically according to the distance. Chen et al. studied the interaction efficiency of mobile robot in human traffic area to guide future design of robot-assisted pedestrian evacuation algorithms.²² In their research, a laser range finder was integrated in Adept Pioneer P3-DX mobile robot, and a social force model experiment was also applied. The results showed that robot can detect pedestrian and avoiding the collision while maintaining desired average speed. Other obstacle avoidance methods used in path planning of mobile robotics include obstacle-dependent Gaussian potential field.²³ This method is applicable in static and dynamic environment without consideration of acceleration or speed of mobile robot. Distance histogram Bug algorithm is based on average path length and mainly

Table 1. Summary of few path planning and obstacle avoidance methods.

Method	Advantage	Disadvantage
Probabilistic roadmaps	High-speed, parallel implementation. ⁴²	Path search for every individual query. ¹⁹
Simulated annealing	Deals with highly nonlinear models, chaotic and noisy data, and many constraints and can generate smooth path. ⁴³	Very slow when cost function is expensive, cannot tell whether it has found an optimal solution. ⁴⁴
Neural networks	Suitable for artificial intelligence, can be trained to solve nonlinear static and dynamic without knowledge of environment structure. ⁴⁵	Computationally efficient but depend on initial value. ⁴⁶ In 3D environments, the computation complexity will explode simultaneously. ⁴⁷
Genetic algorithms	Better optimality robust based on stochastic search. ⁴⁸	Poor local optimizing, oscillation, and convergence speed largely depends on the encoding method. ^{49,14}
Ant colony optimization	Proved to be able to deal with multiobjective path planning problems, and it is also able to tackle continuous planning problem. ⁵⁰	The efficiency yet needs to be tested in real physical experiment. ⁵¹
Particle swarm optimization	Simple to implement and requires less computing time and performs well in various optimization problems. ⁵²	Easy to fall into local optimum in high-dimensional space and has a low convergence rate in the iterative process. ⁵³
Fuzzy logic	Ability to make inferences even in uncertain scenarios with static and dynamic obstacles. ^{54,55}	Robot stops to update the environmental map to determine the next move. ⁵⁶

implemented for static environment.²⁴ Other emerging techniques such as neural network, Fuzzy logic, and genetic algorithms are used for complex situations such as uncertain workspace and multiple inputs/outputs.^{25,26}

Safety methods for human–robot interaction in a fenceless robotic cell have been investigated with effort to maximize admissible velocity of a robot and to reduce injury. The projection of human arm motion into robot path was investigated to estimate the collision of robot.²⁷ This method is based on the optimal path related to the shortest duration of movement using segmentation of path and nonlinear programming; however, it does not consider size of objects and how speed control can be implemented to improve safety. Moreover, it uses noncost effective motion-tracking system that might be impractical for industrial application. Polverini et al. introduced assessment method using Kinetostatic safety field index only to study real-time collision avoidance in unstructured and dynamic environments.²⁸ The method uses mesh technique to calculate the index based on relative position and velocity vectors in real time between a moving rigid body “source of danger” and a point in space. Ragaflia et al. introduced trajectory generation algorithm acquired from different depth sensors for estimation of the kinematic configuration of human worker moving inside the robotic cell to maximize productivity function subjected to safety constraint.²⁹ The proposed algorithm takes into account volumes as obstacles modify the path of preprogrammed task of robotic arm. Sharkawy et al. utilized torque sensors data in multilayer feedforward neural network to train the dynamic model of joint manipulator with and without external contacts.³⁰ Mainprice and Berenson presented a framework that allows the human and the robot to perform simultaneous manipulation safely based on early prediction of the human motion to enable collaborative tasks in close

proximity.³¹ Wang et al. introduced a safety index with a distance factor is defined to describe the level of safety during physical human–robot interaction. The optimization problem was solved within the sampling time of the system allowing algorithm be applied in real-time trajectory generation.^{32,33} Kanazawa et al. obtained robot’s trajectory based on a probabilistic prediction of the worker’s motion and the reduction of waste time simultaneously.³⁴ Notable summary of advantage and disadvantage of popular obstacle avoidance method are presented in Table 1. More recent in-depth reviews of techniques related to the navigation of unmanned mobile vehicles through various environments with obstacles can be found in the following literatures.^{35,36,37,38,39,40,41}

This article presents the development of obstacle avoidance method for industrial robots in manufacturing workplace based on prediction of dynamic motion of an approaching worker or robot within a shared workplace. Furthermore, the article discusses a mechanism to maximize efficiency and safety of shared workspace. The article introduces real-time prediction model of collision between a stationary industrial robot and a moving target based on a cost function that accounts for estimated time of impact, distance to impact, and the affected area of collision. The article offers a novel continuous speed control of robot based on simple cost function. Unlike majority of traditional industry-based safety methods which relies on the events to take place or on enforcing robot to change its path, the goal is to warrant robot or worker enough time by slowing down robot’s speed; therefore, giving worker enough time to take corrective action and avoid collision that may result in production downtime and injuries. The article is organized as follows: a description of the robot–worker shared space is defined in the “Related work” section. The relative position derivation is obtained from

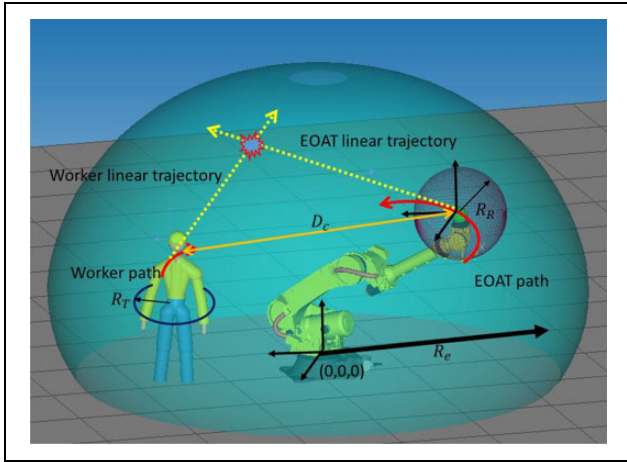


Figure 1. A worker in work envelope of a robot.

distributed sensor measurement. Linear dynamic path prediction of worker is derived and used in collision and safety models. An illustrative simulation example is presented in detail in the “Method” section. Worker trajectory and robot path are used together to calculate collision parameters, which are then used in cost function. Finally, collision and safety model is suggested to calculate the probability of collision and its effect on robot efficiency.

Method

Problem definition

This project investigates a work environment comprised of an industrial robot arm and a fast-moving worker shown in Figure 1. The goal is to track the position vector of the worker relative to the robot's EOAT and then predict whether the motion trajectories would lead to possible collision. A collision cost function that decides the threat level is defined here from three parameters, simultaneously: time-to-impact t_c , the relative distance between the EOAT and worker D_c , and the predicted overlapping area ratio at impact S_c . The t_c is calculated from the time it takes the linear trajectories of both EOAT and the worker to intersect. Therefore, the values of t_c if exist vary from zero to infinity, where collision is more probable to occur if the t_c value is close to zero. Likewise, a large value of t_c detected within a small shared space refers to an insignificant relative speed measured between EOAT and worker. A false alarm could be generated in occasions when the value of D_c is small but the relative speed is very small. In such scenario, it is important to introduce other indicators such as magnitude of overlapping between the two trajectories calculated at different times. Because the position vector of worker could be calculated in reference to the EOAT by using geometrical transformation, it becomes possible to define a dynamic safety area around robot's EOAT, say a circle or a square with

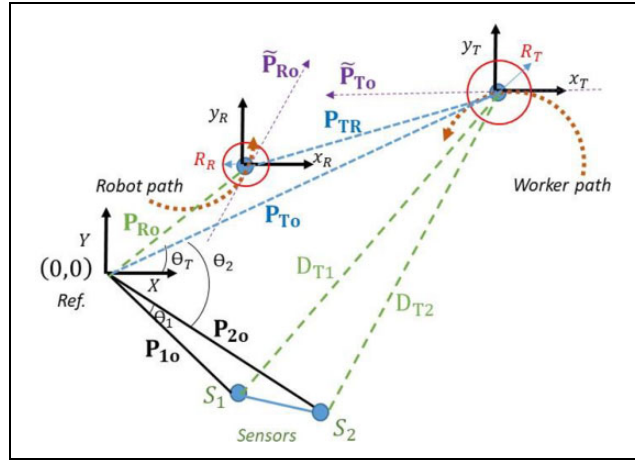


Figure 2. Kinematic model for tracking worker. The position vector \mathbf{P} is represented in Cartesian coordinate system with components $\mathbf{x}\mathbf{i} + \mathbf{y}\mathbf{j}$.

characteristic length of R_R , within which interference with worker would be restricted. Therefore, this allows selecting an area that encloses dangerous part of robot arm. Similarly, worker area is defined by characteristic length R_T , and work envelope Ω_e of a robot is defined by characteristic length R_e . In later section, an energy-based cost function is utilized to combine the aforementioned parameters. The value of such cost function could be evaluated in real time to evaluate the threat such as no intersection and worker is far-off robot's EOAT, that is, ($t_c = \{\emptyset\}$, $D_c \gg R_R$, or $D_c > R_e$), or a worker is marginally within work envelop of robot but the relative speed is small, that is, ($t_c \gg 0$, $R_e > D_c > R_R$), or speed of worker is fast relative to robot and marginally close ($t_c \rightarrow 0$, $R_e > D_c > R_R$), or no intersection and worker is marginally close, that is, ($t_c = \{\emptyset\}$, $R_e > D_c > R_R$), and many other combinations. Under this definition, the threat level is customizable to the safety and production requirements. Moreover, the location of the restricted area is dynamic and became a subset of the working envelops of the robot. Unlike majority of traditional methods which have deployed static zones around robot all times no matter where the EOAT is located, or in most ongoing research proposal for a zone with dynamic radius, our proposed method promote to utilize space effectively by alarming system around collision areas. Ultimately, this facilitates the installation large number of robots in a workplace.

Measurement of the relative position

A simplified two-dimensional (2D) kinematic model for the work environment in Figure 1 is obtained and shown in Figure 2. The dynamical model assumes two moving areas in planner workspace with the goal to better understand the trade-offs between safety and efficiency, while a 3D space model can be simply obtained by adding the third

independent motion into the model in terms of depth coordinate. An arrangement of at least two position sensors S_1 and S_2 would provide 2D measurements within a confined area. The sensors are fixed relative to a world Cartesian coordinate at absolute position vectors P_{1o} and P_{2o} , respectively. Network of sensors (such as laser, ultrasonic, and cameras) could be distributed and fused to measure larger area. In this research, the derivation presented for pair of sensors. A bigger area can be covered by adding replicas of this model all around the robot where each pair of sensor covering a segment. The absolute position vector of the EOAT P_{Ro} is recorded by the robot's encoders in real time t . The absolute position vector of the worker P_{To} is indirectly obtained by two linear distance measurements D_{T1} and D_{T2} . The position vector of the worker relative to the EOAT is P_{TR} and equals to $P_{To} - P_{Ro}$. The derivation of the position vector of the worker follows the trigonometry in Figure 2.

The angle formed by vectors P_{1o} and P_{2o} is simply given by

$$\theta_1 = \tan^{-1}\left(\frac{y_{1o}}{x_{1o}}\right) - \tan^{-1}\left(\frac{y_{2o}}{x_{2o}}\right) \quad (1)$$

Apply the cosine law for each of the two triangles formed by the origin of the world Cartesian coordinate system, the worker and each sensor. This yields

$$D_{T1}^2 = \|P_{To}\|^2 + \|P_{1o}\|^2 - 2\|P_{To}\|\|P_{1o}\|\cos(\theta_1 + \theta_2) \quad (2)$$

$$D_{T2}^2 = \|P_{To}\|^2 + \|P_{2o}\|^2 - 2\|P_{To}\|\|P_{2o}\|\cos\theta_2 \quad (3)$$

The unknown distance $\|P_{To}\|$ and the angle θ_2 are obtained numerically by solving the two nonlinear equations simultaneously.

Then calculate the worker angle θ_T measured CCW from X-axis by using

$$\theta_T = \theta_2 - \tan^{-1}\left(\frac{y_{2o}}{x_{2o}}\right) \quad (4)$$

Finally, the position components of the worker relative to the world Cartesian coordinate are obtained as follows

$$(P_{To})_x = \|P_{To}\|\cos\theta_T, \quad (P_{To})_y = \|P_{To}\|\sin\theta_T \quad (5)$$

In summary, it is plausible to calculate the 2D position of the worker by using two noncollinear distance sensors. The accuracy of the worker's position is sensitive to the quality of the data measured in real time. Therefore, high-end sensors (i.e. capable of extracting depth such as ultrasonic, laser, radar, optical camera, and motion capture system) are recommended together with classification and filtration techniques that help reduce undesired noises in the prediction model. For example, generalized hough transform technique could be used in vision systems to extract the key features of moving object including its dimension and distance.⁵⁷

Linear dynamic trajectory

Consider the worker position P_{To} is measured for $t_o - n\Delta t \leq t \leq t_o$, where n is the number of samples, Δt is the sampling time increment of the sensors, and t_o represents the present time or the initial time at which the future path of the worker is unknown. On the other hand, the industrial robot is preprogrammed and its EOAT position P_{Ro} is often registered and given for $\forall t \geq t_o$. Because the worker position is unknown to the robot for $t > t_o$, future collisions between the EOAT and worker cannot be identified unless there exists a trajectory model that helps predicts the future position of the worker \tilde{P}_{To} . The prediction trajectories create positions in area Ω_p , where $\Omega_p \subset \Omega_e$.

In this article, it is assumed that the dynamic motion of the worker at $t = t_o$ is under constant force with constant acceleration vector $\ddot{P}_{To}(t_o)$, and initial velocity vector $\dot{P}_{To}(t_o)$. This assumption could also be applied to the robot in the absence of encoder readings, where the predicted position of EOAT \tilde{P}_{Ro} replaces the true value P_{Ro} . Therefore, the predicted position vector of the worker can be written by using equation of motion at constant acceleration

$$\begin{aligned} \tilde{P}_{To} &= P_{To}(t_o) + \dot{P}_{To}(t_o) \times t + 0.5 \ddot{P}_{To}(t_o) \times t_p^2 \\ &= \tilde{x}_{To}(t_p)i + \tilde{y}_{To}(t_p)j \end{aligned} \quad (6)$$

where the prediction time $t_p = [0, \Delta t_{fp}]$ is simulated until a final time Δt_{fp} with a desired sampling rate Δt_p . The velocity and acceleration are calculated by using backward finite difference method⁵⁸

$$\dot{P}_{To}(t_o) = (P_{To}(t_o) - P_{To}(t_o - \Delta t))/\Delta t \quad (7)$$

$$\ddot{P}_{To}(t_o) = (P_{To}(t_o) - 2P_{To}(t_o - \Delta t) + P_{To}(t_o - 2\Delta t))/\Delta t^2 \quad (8)$$

Therefore, it is sufficient to store three position samples at $t = \{t_o, t_o - \Delta t, t_o - 2\Delta t\}$ to predict the trajectory of the worker. While the rate of change of worker's velocity is assumed to be constant at instant, the model keeps updating the value of acceleration regardless of its magnitude. It shall be noted here that the proposed trajectory model is also deployable in case of robot-robot interaction and might produce more accurate results specially because human motion could be based on random walk.

It should be noted that the trajectory in equation (6) provides a realistic assumption about the motion of the worker. This could be justified because the trajectory is periodically updated. Other models relevant to social behavior could be investigated such as random walk, Markov Chain, and swarm or Brownian motion. Moreover, the linear trajectory model should be sufficient to predict the motion of worker when sampling rate is increased; however, the risk of collision is expected to be higher.

Collision model

A collision occurs in an areas Ω_c at time set $t_c = \{t_1, \dots, t_k\}$ and when the predicted path of the worker $\tilde{P}_{To}(t)$ and the true path of the EOAT $P_{Ro}(t)$ intersect inside the workplace envelope, that is, $\Omega_c \subset \Omega_e$. The worker and the EOAT are defined here by 2-D areas such as a square or circular shape. For example, the parametric functions defining the position of the circular worker area Ω_T are computed for $\forall \omega$ and $\forall r_T$

$$\begin{aligned} x_{T_A}(t) &= \tilde{x}_{T_o}(t) + r_T \cos \omega \\ y_{T_A}(t) &= \tilde{y}_{T_o}(t) + r_T \sin \omega \end{aligned} \quad (9)$$

where $\omega \in [0, 2\pi]$ and $r_T \in [0, R_T]$. Similarly, the parametric functions defining the position of the circular moving area Ω_R of the EOAT are computed for $\forall \omega$ and $\forall r_R$

$$\begin{aligned} x_{R_A}(t) &= x_{R_o}(t) + r_R \cos \omega \\ y_{R_A}(t) &= y_{R_o}(t) + r_R \sin \omega \end{aligned} \quad (10)$$

where $r_R \in [0, R_R]$. Regardless of the shape of the areas representing the robot or worker, the collision occurs when the following criterion holds

$$\begin{aligned} (x_{R_A}(t), y_{R_o}(t)) \cap (x_{T_A}(t), y_{T_A}(t)) &\neq \Phi \\ \Delta t_{rf} \geq t \geq t_o \end{aligned} \quad (11)$$

where Δt_{rf} is the final real-time simulation used to study the robot and worker movement. The solution of the criteria provides a time set for collisions t_c with corresponding locations. Therefore, the time-to-impact is defined here by the time it takes a worker or a robot to respond for such threat is $t_c - t_o$. In addition, the overlapping area ratio S_c at impact t_c is a useful measure of safety and it is defined here by the ratio of the overlapping areas to the total areas, that is, $(S_c = \Omega_c \cap \Omega_R) / (\Omega_c \cup \Omega_R)$. All of these predictive measures with the true distance between the centers of the worker and EOAT at t_o , that is, $D_c = \|P_{TR}(t_o)\|$ are independent and can be used to develop strategies for mitigation the risk of collision. It should be noted here that those control variables comprise the collision model M_c and must be all updated in real time as the position of the worker changes. Also, the computational efficiency required to solving equation (11) can be substantially improved by narrowing down the search window within the area range covered by sensors.

Safety model

The goal of a safeguard system is to continuously control the velocity $\dot{P}_{Ro}(t_i)$ of the robot at any sampled time t_i , depending on the predicted level of collision. The severity of the collision is obtained from the safety control model $M_S(t_i)$ in Figure 3, which uses collision model variables as control inputs: (1) inverse of time-to-impact $x_1(t_i) = 1/(t_c - t_o + \delta)_i$, (2) inverse of the distance between robot

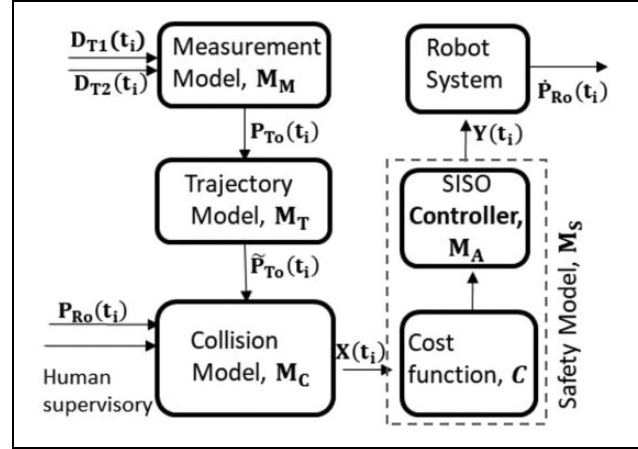


Figure 3. Suggested safeguard control system of a worker within the vicinity of an industrial robot.

and worker $x_2(t_i) = 1/(\|P_{TR}(t_o) + \epsilon\|)_i$, (3) the affected or overlapping areas ratio at collision $x_3(t_i) = (S_c)_i$, (4) an auxiliary or expert input $x_4(t_i)$, and (5) prior history of control input $X = [x_1, x_2, x_3, x_4]$.

The constant values ϵ and δ are small real numbers of order magnitude between $\{-1$ to $-4\}$ used to bound the value of the cost function from reaching infinity when the relative distance or the time to collision are zero. The purpose of x_4 is to supervise or adjust the response to desired standard practices. Therefore, the worker is considered to be safe when all the state variables go to zero, that is, $x_1 \rightarrow 0$ when $(t_c - t_o) \rightarrow \infty$, and $x_3 \rightarrow 0$ when $S_c \rightarrow 0$, $x_1 \rightarrow 0$ when $(t_c - t_o) \rightarrow \infty$. The overlapping area ratio—if exists—is bounded $S_c \in [0, 1]$. According to the definition of the overlapping area ratio, its maximum value S_{cmax} can be achieved when the worker area Ω_T is completely inside the robot area Ω_R or vice versa. Because the areas are already known, their values can be calculated as follows: $\max(\Omega_T \cap \Omega_R) = \Omega_T / (\Omega_T + \Omega_R)$ if $\Omega_T \leq \Omega_R$ or $\max(\Omega_T \cap \Omega_R) = \Omega_R / (\Omega_T + \Omega_R)$ if $\Omega_R \leq \Omega_T$. While there are many unaccounted variables that effect the safety, it is crucial to design a safety model that controls the speed of the robot using input vector X . Figure 3 suggests a safety model comprised of cost function C and SISO controller M_A . It is desired to produce a bounded output Y that is not only proportional to the cost function but also can be used to easily override the speed of the robot. For example, consider safety model M_S be comprised of multiplication of $C = XWX^T$ and attenuator $M_A = 1/A$, where W is 4×4 matrix and A is real number. The cost function is studied for a desired collision area $\Omega_m \subset \Omega_e$. The safety model outputs become bounded quantity $1 \geq Y \geq 0$. The upper values of Y indicate that the robot can be operating close to the maximum speed, and the lowest values imply that the robot speed should be slowed down.

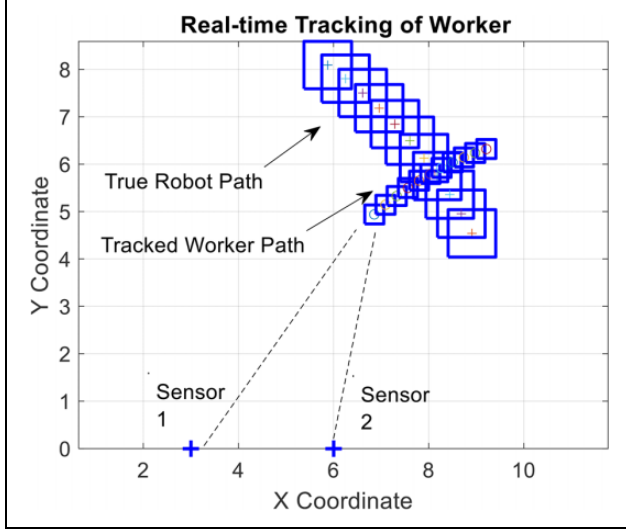


Figure 4. True paths of the worker and the robot simulated over time period $t = [\pi/2, \pi]$.

Illustrative simulations

Real-time tracking and calculation of true impact

In a real-time prediction model, a discrete-step solver for simulation could be chosen for robot controller. Most modern robot controllers are running a cyclic control software, for example, 10 ms (or the better one 1 ms) for the position control loop. KUKA, for example, provides an RSI interface running at 1 ms and FANUC has a similar system called “dynamic path modification.” Furthermore, vision system suffers from delay time during tracking of an object. Vision systems typically do not operate at 1 kHz rather most vision system run at 10–50 frames/s. In this simulation, long-range laser triangulation distance sensor is assumed, which typically has measurement rate much better than 1 kHz.

In this example, a robot moves in circular coordinate with parametric functions described by $(P_{Ro})_x = 10\cos(0.3t)$ and $(P_{Ro})_y = 10\sin(0.3t)$ and it is simulated for period of time $t = [\pi/2, \pi]$. The position of the worker is measured indirectly by two linear distance sensors at sampling rate $\Delta t = \pi/20$. The sensor measurements are simulated where sensor 1 reads $D_{T1} = 5\sqrt{t}$ and sensor 2 reads $D_{T2} = 4\sqrt{t}$. The position components of the worker relative to the world coordinate are reversed engineered by applying equations (1–5). The distance $\|P_{To}\|$ and the angle θ_2 are solved for every time step using the “fsolve” function in MATLAB software.⁵⁹ Then the worker $((P_{To})_x, (P_{To})_y)$ positions are computed and plot with robot position $((P_{Ro})_x, (P_{Ro})_y)$ as shown in Figure 4. The robot area Ω_R and the worker area Ω_T are moving in spatial domain and simulated by square shapes whose lengths are $r_R = 1$ and $r_T = 0.4$, respectively.

Table 2. True impact between worker and robot

$t_c - t_o^a$	S_c^b	$(P_{To})_x$	$(P_{To})_y$	$\ P_{TR}(t_c)\ $
2.1991	0.0164	7.7987	5.6525	0.4876

^a t_o is zero when the algorithm is used to check true collision.

^bCalculated in MATLAB using “bboxoverlapratio” function.

A true or “physical” impact between the worker and the robot is said to occur when the true moving areas physically overlap, that is, the impact time $t_c \neq \{\emptyset\}$. The true impact time is obtained by replacing $\hat{x}_{T_o}(t)$ with $(P_{To})_x$ and $\hat{y}_{T_o}(t)$ with $(P_{To})_y$ in equation (9) and then substituting the latter values in equation (11). The case study is studied for period of time $t = [\pi/2, \pi]$ at sampling rate $\Delta t_p = \Delta t$. The overlapping condition is checked at every time step, and the collision variables are recorded in Table 2. The results show that the true impact time took place at $t_c = 2.1991$ where the worker absolute position is $P_{To} = 9.6317$. The percentage of the overlapped areas is $S_c = 1.64\%$ at a distance of $\|P_{TR}\| = 0.4876$ calculated between the center points of the worker and the robot. The results presented in Table 2 assume no prediction model enabled. The next section computes future collision from dynamic prediction equation of the worker.

Predictive impact model

Real-time tracking provides current and prior information for the position of the worker as function of time, but future positions are unknown due to the randomness of the human intention. Therefore, majority of safeguard technologies used in industrial robotics environment rely on the actual presence of the worker, where robot’s controller is alerted only when the worker resides within predefined zones that are fixed around the space of the robot. Such alerts might be too late to prevent accidents especially when the worker or the robot movement is too fast! Instead, a time-varying predictive model of the worker path was introduced in equation (6) to provide a set of future positions of the worker calculated at every instantaneous position of the robot. This simply appears as trajectories tangent to the path of worker.

In the following simulation example, the case study in the previous section is used to predict the trajectory profile of the worker by using equation (6–9) starting at absolute time instant $t = t_o = 6\pi/10 = 1.885$. The profile is generated for a period of time $\Delta t_f = 2$ and plot in Figure 5. Also, the true path of the robot is simultaneously simulated using given parametric functions. Criterion in equation (11) is evaluated at time t_o and with worker trajectory generated for future time period $t_p = [0, 2]$. Table 3 provides a summary of the simulation results and shows that the robot center is 1.1162 apart from worker center at time $t = t_o = 1.885$ with estimated time-to-impact is 0.314. The percentage of area overlapped at impact moment is 2.4% with 0.4766 apart from each other’s center points.

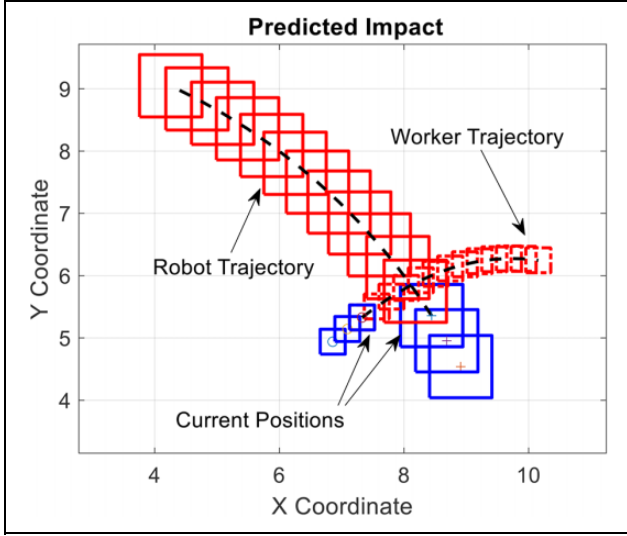


Figure 5. The simulation of the predicted trajectory of the worker starts at $t_o = 6\pi/10$ for time period of $\Delta t_f = 2$ and with sampling rate of $\Delta t_p = \pi/20$. The path of the robot is generated for simultaneously using given parametric functions. The first three sampled steps are not included in prediction.

Table 3. Predicted impact between worker and robot studied at time instance $t_o = \pi/2 + \pi/10$.

t_o	$t_c - t_o$	S_c^a	$(P_{To})_x$	$(P_{To})_y$	$\ P_{TR}(t_o)\ $	$\ P_{TR}(t_c)\ $
1.885	0.314	0.024	7.798	5.664	1.1162	0.4766

Table 4. Predicted impact between worker and robot studied for time period $\forall t_o = [\pi/2, \pi]$ at step size $\Delta t = \pi/20$.

t_o	$t_c - t_o$	S_c^a	$(P_{To})_x$	$(P_{To})_y$	$\ P_{TR}(t_o)\ $	$\ P_{TR}(t_c)\ $
1.885	0.314	0.024	7.798	5.664	1.1162	0.4766
2.042	0.157	0.021	7.799	5.659	0.6678	0.4810
2.199	0	0.016	7.799	5.653	0.4876	0.4876

The above procedures are repeated for every time step t_o in the time range $t = [\pi/2, \pi]$ and with time step increment of $\Delta t = \pi/20$. The first three increments are ignored because the prediction model requires prior knowledge of the past three positions. The results in Table 4 show prediction of two impacts estimated at times $t_o = \{1.885, 2.042\}$ and with time-to-impacts are $\{0.314, 0.157\}$, respectively. The third predicted collision coincides with true impact where $t_o = \{2.199\}$ and time-to-impact is zero.

Cost function and the probability of collision

Let the multi-input-single-output model $M_S = XW X^T / A$ be bounded in a spatial domain defined by the positions of a worker traveling along predicted paths shown in Figure 6 and with overall area defined by $\Omega_m = \Omega_p$. The design of

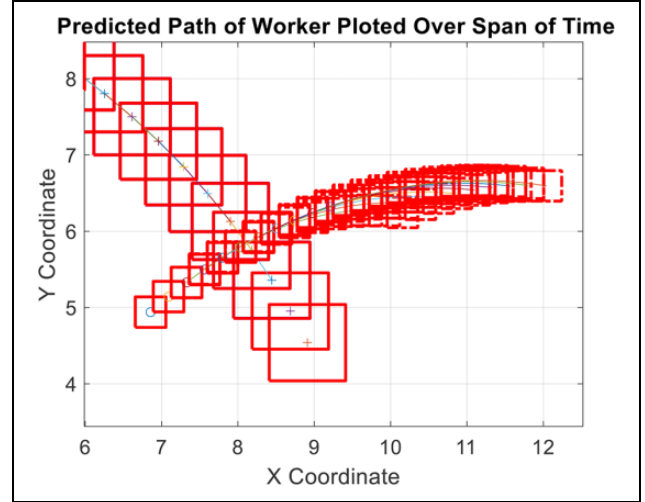


Figure 6. Simulation of the predicted trajectories of the worker and the true path of robot generated over $t = [\pi/2, \pi]$.

the coefficients in the weight matrix of the cost function should be such that it produces maximum values when the true collision condition is satisfied. The cost function should also provide an indication of collision severity proportional to the value of the maximum values. According to the simulation results in Table 4, there are three possible future collisions, with one being a real impact at $(t_c - t_o) \approx 0$. We want to inspect a diagonal weight matrix for the cost function C . This eliminates cross-correlation between the state variables and assumes orthogonality, that is, independent from each other. The following cost function is suggested

$$C(t_o) = \omega_1 x_1^2 + \omega_2 x_2^2 + \omega_3 x_3^2 + \omega_4 x_4^2 \quad (12)$$

where ω is the coefficient corresponding to the state variable x . This cost function is computed for $\forall t_o$. In this special case study, it is assumed that there is no external control signal, that is, $\omega_4 = 0$. The values of x_1 and x_3 are zero when there is no predicted collision. However, $0 < x_3 < S_{cmax}$ when a collision is predicted.

In this simulation example, we choose $\omega_1 = \omega_2 = 1$ and $\omega_3 = \alpha \gg 1$, where α is a correction gain used to amplify the effect of the overlapping term in the cost function. Let $\epsilon = \Delta t_p/2 = 0.0785$, $\delta = 0.001$ and $\alpha = 100$. The cost function is evaluated for a range of time $t = [\pi/2, \pi]$ with step size $\Delta t = \pi/20$. In general, when the cost function value is $C \gg 0$, a collision is expected, while it becomes safe when the value $C \rightarrow 0$ as shown in Figure 7. More particularly, the set of solutions in Table 4 generates large values and with the cost function of the true impact being the maximum. The cost function response in Figure 7 has Gaussian distribution with one peak at the time of true impact $t_o = \{2.199\}$. The positive value of the slope dC/dt within left side of the bell shape in Figure 7—time range $[\pi/2, 2.199]$ —indicates an increasing risk of collision as the time progresses. This is because the worker

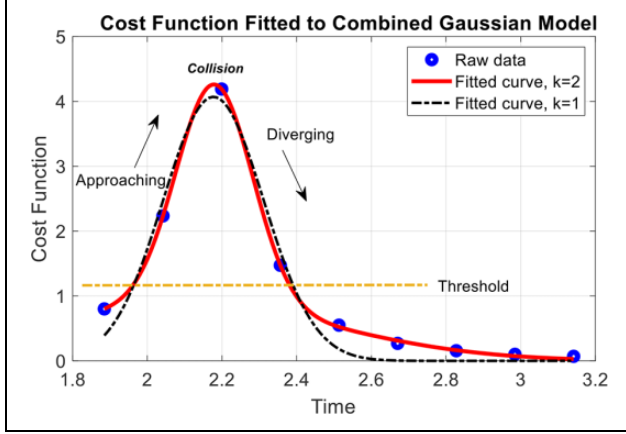


Figure 7. A fitted cost function curve simulated over $t = [\pi/2, \pi]$.

is approaching the true impact condition. However, the negative slope in right side of bell shape—time range $[2.199, \pi]$ —means that the worker is moving away from collision and therefore become safer. Moreover, one could set a threshold value based on Table 4 and above which all the predicted and true collisions would occur.

The cost function data in Figure 7 could be fitted into a combined Gaussian curve of form

$$\check{C}(t) = \sum_{n=1}^k a_n e^{-\left(\frac{t-\mu_n}{\sigma_n}\right)^2} \quad (13)$$

where k is the number of Gaussian functions. The second-order Gaussian, that is, $k = 2$ can be obtained by using MATLAB statistical toolbox “fit” function with “guess2” type, which yields

$$\check{C}(t) = 3.41e^{-\left(\frac{t-2.18}{0.1447}\right)^2} + 0.87e^{-\left(\frac{t-2.1}{0.5649}\right)^2} \quad (14)$$

The above model provides accurate fitting of cost function at any time. To better understand the meaning of the distribution function, a first-order Gaussian distribution is generated using “guess1” function in MATLAB and then plotted together with the second-order and the raw data in Figure 7. The fitted first-order Gaussian function with $k = 1$ yields

$$\check{C}(t) = 4.07e^{-\left(\frac{t-2.177}{0.1909}\right)^2} \quad (15)$$

This first-order Gaussian distribution reveals that the true impact happens at the mean value $\mu = 2.177$ with standard deviation $\sigma = 0.1909$. The curve follows the normal distribution where the cumulative distribution function could be obtained by integrating equation (15). The “normcdf” function in MATLAB software calculates the cumulative distribution function evaluated at time t . Figure 8 is a plot of the collision probability $P(t)$ evaluated over time range $[-\infty, t]$. Therefore, the probability of a

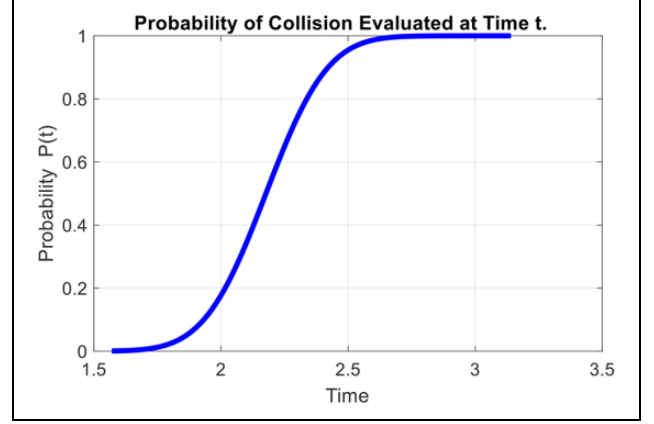


Figure 8. Probability of collision curve simulated over $t = [\pi/2, \pi]$.

collision in time range $[t_1, t_2]$, that is, $t_2 \geq P(t) \geq t_1$ is equal to $P(t_2) - P(t_1)$. Alternatively, the probability of collision can be obtained by subtracting areas under the curve of Figure 8. The figure indicates that the collision is more likely to occur as human approach time μ .

Higher-order Gaussian combinations are suggested for scenarios involving multiple true impacts which appear in the form of peaks in the cost function curve. For example, $k = 4$ could provide a good fit for two true impacts. However, other forms of cost function, different from quadratic form proposed in equation(12), could produce non-Gaussian distribution.

Safety control system

The controller model selected in this example is a simple attenuator $1/A$ where the value $A = \max. C$ is calculated for all sum of areas navigated by the worker and the robot. In the previous example, the total area, shown in Figure 6, has minimum attenuation of $1/A = 0.24$. The value of $\check{C}(t)/A$ normalizes the safety model output to a scale of 100%. This guarantees that the output of the controller model is proportional to the risk of collision and, therefore, can be directly used to control the speed of the robot $v_{Ro}(t)$. One could program the robot’s joint or axial speed to $v_{Ro}^*(t) = (1 - C(t)/A)v_{Ro}(t)$. In the example studied previously, the robot moves in circular coordinate with constant speed $v_{Ro} = \|(\dot{P}_{Ro})_x + (\dot{P}_{Ro})_y\| = 3$. The speed of the robot corresponding to the threat posed by the worker movement is plot in Figure 9. As expected, the speed of the robot winds down as the worker approaches dangerous conditions, comes to full stop at collision moment, and then gains speed quickly as the worker moves away. It shall be noted here that the original path of robot is kept unchanged and the collision avoidance is presented in the sense of zero contact force at contact between the colliding objects.

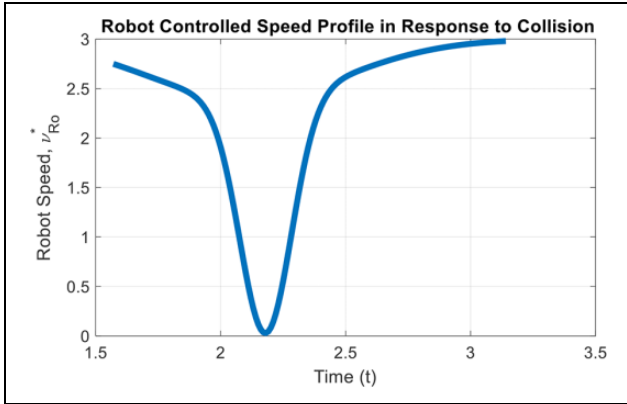


Figure 9. Speed of robot after applying safety control system.

Robot productivity

Assume the robot productivity index is measured by the total energy cost function computed over a range of time

$$Prod = \frac{1}{2} \int v_{Ro}^*(t) dt \quad (16)$$

When there is no worker around the robot, the robot speed remains unchanged. In the example discussed earlier, an uncontrolled robot operates at normal speed of $v_{Ro} = 3$, where its total energy over time $[\pi/2, \pi]$ is computed by using equation (16). The productivity index is equal to $3\pi/4$. However, productivity index of a robot under controlled speed can be computed from the area under the curve in Figure 9 or by numerically integrating the productivity function $Prod = 1.5(1 - C(t)/0.24)$, with $\tilde{C}(t)$ being the function obtained in equation (14). The value of the productivity index under a controlled speed—the worker is present and the controller is enabled—being 1.778. The overall reduction in the robot productivity as results of applying the proposed safety control system is only %24.5. Moreover, another alternative productivity index analogous to energy function could be obtained by raising the power of the robot's speed in equation (16) by the value of 2.

Discussion

The collaborative human–robot interaction introduced in this research acquires information from the pair of sensors to update the robot Euclidean position in real time. In an unknown manufacturing environment, where worker's motion trajectory is uncertain, an auxiliary microprocessor predicts future positions of the worker by assuming a worker moving in a constant acceleration. Because the proposed method uses minimum number of sensors, high sensitive and accurate distance sensors should be installed to improve the reliability of the calculated safety level.

The trajectory generated by the microprocessor is kept updated in real time. On the other hand, because the robot path is typically preprogrammed and deterministic, it is

compared with the generated trajectories in real time with goal to search for possible future intersection(s). The simplicity of the mathematical model made it easy to impose multiobjective functions and control scheme. This prediction model was then combined with manufacturing efficiency through a contentiously updated cost function which is directed to control the speed of the robot. The key approach for predicting a collision is pertained by applying four factors that encapsulate the predicted time to collision, distance to collision, overlapping area at collision, and a manual remote control parameter determined by the user.

The simulated case study produced a cost function with multimodal Gaussian distribution. The mean values of the distribution are located at the peaks and correspond to high probability of collision. At such instants, the robot smoothly decelerates to a quasi-static stop and then accelerates to continue its normal operation specially when the threat is sufficiently far-off. The smaller the variance of the distribution is, the less time is given to the robot to react to the threat. However and in accordance to the proposed productivity index—which was defined by area under speed and time curve—the robot efficiency increases when the gradient of the cost function is steep.

The physical realization of the proposed prediction method becomes feasible to implement mainly because it requires few measurements as compared to classical and modern obstacle avoidance methods. These data are collected from a pair of sensors configured to scan a sectional area whose size depends on the range of sensing technology used. The method can be extended to cover larger area by summing data from other pairs of distributed sensors.

Future work

The preliminary setup in Figure 10 is being constructed to deploy the developed safety algorithm in an open workspace environment where a threat—like a worker—is prominent. The system is comprised of Mitsubishi RH-6FHD robot with CR-750 controller, and two low-cost long-range ultrasonic sensors connected to an external Arduino controller. The worker position and the cost function are being calculated in real time using the Arduino microprocessor, which then sends signal to the robot I/O. Currently, the Arduino controller is programmed to produce three digital outputs, and a robot program—written in MELFA RT-toolbox software—operates at three distinct speeds depending on the value received from I/O. However, this setup requires preprogramming and synchronizing with the motion of the EOAT, which is impractical for real-time experiment and verification. Therefore, we plan to continue with development by enabling Arduino and robot controllers to talk back to each other using analog signal instead of digital. More particularly communicate the position of the EOAT to PC and have it scanned by Arduino via USB port.

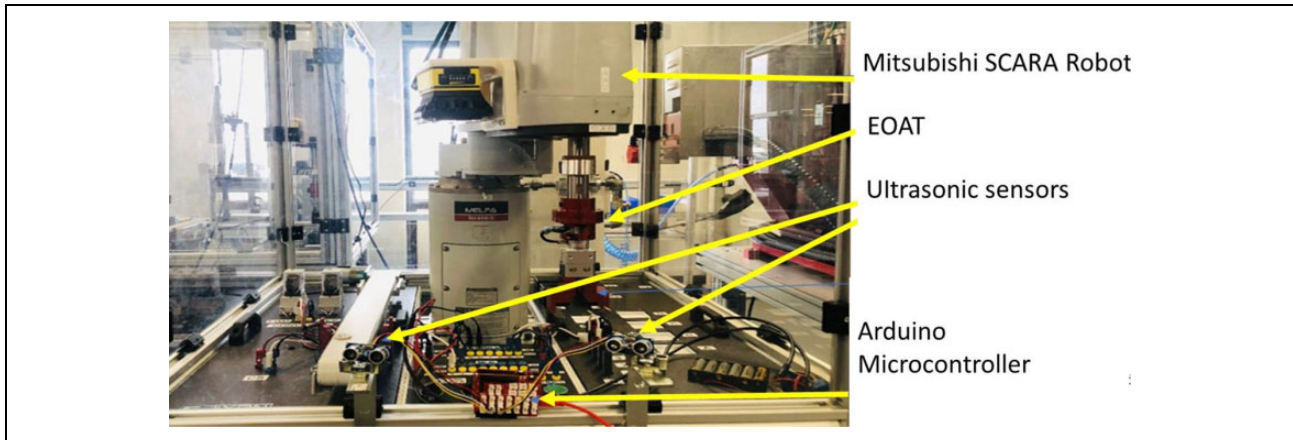


Figure 10. Experimental apparatus.

Conclusion

We developed a continuous speed control system to improve human–robot safety while robot productivity is not compromised. The predictive dynamic motion model for a human moving relative to a robot provided an accurate mean to estimate the future impact conditions which are used to construct a Gaussian cost function. This is used to calculate the probability of collision within a range of time and design a safety model to control speed of the robot in real time and based on the overall threat level. The case study showed that for an EOAT of a robot moving in deterministic motion under no safety control, an arbitrarily approaching worker poses a threat that would lead to a physical impact. However, when the safety control was deployed, the robot's speed responded proportional to the cost function and led to only 24% reduction in productivity during entire operation. This is very promising result specially because it does not lead to complete downtime in robot operation. The proposed algorithms can be used in application involving object locomotion such as mobile robots, driverless cars, and unmanned aerial vehicles.



Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) disclosed receipt of the following financial support for the research, authorship, and/or publication of this article: The research is conducted under support from Ohio Department of Higher Education (ODHE) and Regionally Aligned Priorities in Delivering Skills Program (RAPIDS 4.0) grant “Automated Guided Vehicles for Smart-logistics and Safe Material Handling in Advanced Manufacturing and Warehouse Industries.

ORCID iD

Mohammad Mayyas  <https://orcid.org/0000-0002-8059-8297>
 Muhammed A Syed  <https://orcid.org/0000-0002-2854-3491>

References

1. Hagele M. Double-digit growth highlights a boom in robotics [industrial activities]. *IEEE Robot Autom Mag* 2017; 24(1): 12–14. IEEE.
2. Dhillon BS and Fashandi ARM. Safety and reliability assessment techniques in robotics. *Robotica* 1997; 15(6): 701–708. Cambridge University Press.
3. Murashov V, Hearl F, and Howard J. Working safely with robot workers: recommendations for the new workplace. *J Occup Environ Hyg* 2016; 13(3): D61–D71. Taylor & Francis.
4. Chinniah Y, Aucourt B, and Bourbonnière R. Safety of industrial machinery in reduced risk conditions. *Saf Sci* 2017; 93: 152–161. Elsevier.
5. Byler C, Kesly L, Richardson S, et al. Work-related fatal motor vehicle traffic crashes: Matching of 2010 data from the Census of Fatal Occupational Injuries and the Fatality Analysis Reporting System. *Accident Analysis & Prevention* 2016; 92: 97–106.
6. Bellamy LJ, Alea BJM, Geyer TAW, et al. Storybuilder—a tool for the analysis of accident reports. *Reliab Eng Syst Saf* 2007; 92(6): 735–744. Elsevier.
7. Chinniah Y. Analysis and prevention of serious and fatal accidents related to moving parts of machinery. *Saf Sci* 2015; 75: 163–173.
8. Vogel C, Walter C, and Elkmann N. Safeguarding and supporting future human-robot cooperative manufacturing processes by a projection-and camera-based technology. *Procedia Manuf* 2017; 11: 39–46. Elsevier.
9. Zlatanski M, Sommer P, Zurfluh F, et al. Radar sensor for fenceless machine guarding and collaborative robotics. In: *2018 IEEE International Conference on Intelligence and Safety for Robotics (ISR)*, Shenyang, China, 24–27 August 2018, pp. 19–25.
10. Vysocky A and Novak P. Human-Robot collaboration in industry. *MM Sci J* 2016; 9(2): 903–906.
11. Naderan-Tahan M and Manzuri-Shalmani MT. Efficient and safe path planning for a mobile robot using genetic algorithm.

- In: *2009 IEEE Congress on Evolutionary Computation*, Trondheim, Norway, 18–21 May 2009, pp. 2091–2097.
12. Masehian E and Sedighizadeh D. Classic and heuristic approaches in robot motion planning—a chronological review. *World Acad Sci Eng Technol* 2007; 23(5): 101–106.
 13. Solea R and Cernega DC. Obstacle avoidance for trajectory tracking control of wheeled mobile robots. In: Borangiu T, Trentesaux D, and Thomas A (eds) *Service orientation in holonic and multi agent manufacturing and robotics*. Berlin: Springer, 2013, pp. 279–290.
 14. Ayawli BBK, Chellali R, Appiah AY, et al. An overview of nature-inspired, conventional, and hybrid methods of autonomous vehicle path planning. *J Adv Transp* 2018; 2018(1): 1–27. Hindawi, 2018.
 15. Wang Y, Goila A, Shetty R, et al. Obstacle avoidance strategy and implementation for unmanned ground vehicle using LIDAR. *SAE Int J Commercial Veh* 2017; 10(2017-01–0118): 50–55.
 16. LaValle SM, Gonzalez-Banos HH, Becker C, et al. Motion strategies for maintaining visibility of a moving target. In: *Proceedings of international conference on robotics and automation*, Albuquerque, NM, USA, 25–25 April 1997, pp. 731–736.
 17. Ullah I, Ullah F, Ullah Q, et al. Sensor-based robotic model for vehicle accident avoidance. *J Comput Intell Electron Syst* 2012; 1(1): 57–62. American Scientific Publishers.
 18. Khatib O. Real-time obstacle avoidance for manipulators and mobile robots. In: Cox IJ and Wilfong GT (eds) *Autonomous robot vehicles*. Berlin: Springer, 1986, pp. 396–404.
 19. Svestka P and Overmars MH. Coordinated motion planning for multiple car-like robots using probabilistic roadmaps. In: *Proceedings of 1995 IEEE international conference on robotics and automation*, Nagoya, Aichi, Japan, 21–27 May 1995, pp. 1631–1636.
 20. Nikolakis N, Maratos V, and Makris S. A cyber physical system (CPS) approach for safe human-robot collaboration in a shared workplace. *Robot Comput-Integ Manuf* 2019; 56: 233–243.
 21. Long P, Christine C, Chablat DC, et al. An industrial security system for human-robot coexistence. *Ind Robot* 2018. Emerald Publishing Limited.
 22. Chen Z, Jiang C, and Guo Y. Pedestrian-robot interaction experiments in an exit corridor. In: *2018 15th international conference on ubiquitous robots (UR)*, Honolulu, Hawaii, 26–30 June 2018, pp. 29–34.
 23. Cho J-H, Pae D-S, Lim MT, et al. A real-time obstacle avoidance method for autonomous vehicles using an obstacle-dependent Gaussian potential field. *J Adv Transp* 2018; 2018(1): 1–15. Hindawi.
 24. Zhu Y, Zhang T, Song J, et al. A new bug-type navigation algorithm for mobile robots in unknown environments containing moving obstacles. *Ind Robot* 2012; 39(1): 27–39. Emerald Group Publishing Limited.
 25. Samsudin K, Ahmad FA, and Mashohor S. A highly interpretable fuzzy rule base using ordinal structure for obstacle avoidance of mobile robot. *Appl Soft Comput* 2011; 11(2): 1631–1637. Elsevier.
 26. Duguleana M and Mogan G. Neural networks based reinforcement learning for mobile robots obstacle avoidance. *Exp Syst Appl* 2016; 62: 104–115. Elsevier.
 27. Weitschat R and Aschemann H. Safe and efficient human–robot collaboration part II: optimal generalized human-in-the-loop real-time motion generation. *IEEE Robot Autom Lett* 2018; 3(4): 3781–3788. IEEE.
 28. Polverini MP, Zanchettin AM, and Rocco P. A computationally efficient safety assessment for collaborative robotics applications. *Robot Comput-Integ Manuf* 2017; 46: 25–37. Elsevier.
 29. Ragaglia M, Zanchettin AM, and Rocco P. Trajectory generation algorithm for safe human-robot collaboration based on multiple depth sensor measurements. *Mechatronics* 2018; 55: 267–281. Elsevier.
 30. Sharkawy A-N, Koustoumpardis PN, and Aspragathos N. Human–robot collisions detection for safe human–robot interaction using one multi-input–output neural network. *Soft Comput* 2019; 24(9): 6687–6719. Springer.
 31. Mainprice J and Berenson D. Human-robot collaborative manipulation planning using early prediction of human motion. In: *2013 IEEE/RSJ international conference on intelligent robots and systems*, Tokyo, Japan, 3–7 November 2013, pp. 299–306.
 32. Chen Y-L, Wang S-C, and Wang C-A. Study on vehicle safety distance warning system. In: *2008 IEEE international conference on industrial technology*, Chengdu, China, 21–24 April 2008, pp. 1–6.
 33. Wang J, Wu J, and Li Y. The driving safety field based on driver–vehicle–road interactions. *IEEE Trans Intell Transp Syst* 2015; 16(4): 2203–2214. IEEE.
 34. Kanazawa A, Kinugawa J, and Kosuge K. Adaptive motion planning for a collaborative robot based on prediction uncertainty to enhance human safety and work efficiency. *IEEE Trans Robot* 2019; 35(4): 817–832. IEEE.
 35. Hoy M, Matveev AS, and Savkin AV. Algorithms for collision-free navigation of mobile robots in complex cluttered environments: a survey. *Robotica* 2015; 33(3): 463–497. Cambridge University Press.
 36. Patle BK, Babu LG, Pandey A, et al. A review: on path planning strategies for navigation of mobile robot. *Def Technol* 2019; 15(4): 582–606. Elsevier.
 37. Rubio F, Valero F, and Llopis-Albert C. A review of mobile robots: concepts, methods, theoretical framework, and applications. *Int J Adv Robot Syst* 2019; 16(2): 1729881419839596. SAGE Publications Sage UK: London, England.
 38. Wang N, Zeng Y, and Geng J. A brief review on safety strategies of physical human-robot interaction. In: *3rd international conference on intelligent computing and cognitive informatics (ICICCI 2018)*, Mexico City, Mexico, 19–21 December 2018, p. 1015. EDP Sciences.
 39. Aggarwal S and Kumar N. Path planning techniques for unmanned aerial vehicles: a review, solutions, and

- challenges. *Comput Communications* 2020; 149: 270–299. Elsevier.
40. Panda M, Das B, Subudhi B, et al. A comprehensive review of path planning algorithms for autonomous underwater vehicles. *Int J Autom Comput* 2020; 17(3): 1–32. Springer.
 41. Santos LC, et al. Path planning for ground robots in agriculture: a short review. In: *2020 IEEE international conference on autonomous robot systems and competitions (ICARSC)*, Azores, Portugal, 15–17 April 2020, pp. 61–66.
 42. Akinc M, Bekris KE, Chen BY, et al. Probabilistic roadmaps of trees for parallel computation of multiple query roadmaps. In: *Robotics research. The eleventh international symposium*, Siena, Italy, 19–22 October 2003, 2005, pp. 80–89.
 43. Ho Y-J and Liu J-S. Simulated annealing based algorithm for smooth robot path planning with different kinematic constraints. In: *Proceedings of the 2010 ACM symposium on applied computing*, Sierre, Switzerland, March 2010, pp. 1277–1281.
 44. Miao H and Tian Y-C. Robot path planning in dynamic environments using a simulated annealing based approach. In: *2008 10th international conference on control, automation, robotics and vision*, Hanoi, Vietnam, 17–20 December 2008, pp. 1253–1258.
 45. Janglová D. Neural networks in mobile robot motion. *Int J Adv Robot Syst* 2004; 1(1): 2. SAGE Publications Sage UK: London, England.
 46. Kroumov V and Yu J. Neural networks based path planning and navigation of mobile robots. In: Topalov A (ed) *Recent advances in mobile robotics*. London: InTech, 2011, pp. 174–190.
 47. Kassim AA and Kumar BVKV. A neural network architecture for path planning. In: *[Proceedings 1992] IJCNN international joint conference on neural networks*, Baltimore, MD, USA, 7–11 June 1992, pp. 787–792.
 48. Tuncer A and Yildirim M. Dynamic path planning of mobile robots with improved genetic algorithm. *Comput Electr Eng* 2012; 38(6): 1564–1572. Elsevier.
 49. Liang Y and Xu L. Global path planning for mobile robot based genetic algorithm and modified simulated annealing algorithm. In: *Proceedings of the first ACM/SIGEVO summit on genetic and evolutionary computation*, Shanghai, China, June 2009, pp. 303–308.
 50. Dorigo M, Birattari M, and Stutzle T. Ant colony optimization. *IEEE Comput Intell Mag* 2006; 1(4): 28–39. IEEE.
 51. Guan-Zheng TAN, Huan H, and Sloman A. Ant colony system algorithm for real-time globally optimal path planning of mobile robots. *Acta Automatica Sinica* 2007; 33(3): 279–285. Elsevier.
 52. Nasrollahy AZ and Javadi HHS. Using particle swarm optimization for robot path planning in dynamic environments with moving obstacles and target. In: *2009 Third UKSim European symposium on computer modeling and simulation*, Athens, Greece, 25–27 November 2009, pp. 60–65.
 53. Li M, Du W, and Nian F. An adaptive particle swarm optimization algorithm based on directed weighted complex network. *Math Probl Eng* 2014; 2014(2): 1–7. Hindawi.
 54. Ramanathan R. *Handbook of research on architectural trends in service-driven computing*. Pennsylvania: IGI Global, 2014.
 55. Almasri MM, Elleithy KM, and Alajlan AM. Development of efficient obstacle avoidance and line following mobile robot with the integration of fuzzy logic system in static and dynamic environments. In: *2016 IEEE long island systems, applications and technology conference (LISAT)*, Farmingdale, NY, USA, 29 April 2016, pp. 1–6.
 56. Baldoni PD, Yang Y, and Kim S-Y. Development of efficient obstacle avoidance for a mobile robot using fuzzy Petri nets. In: *2016 IEEE 17th international conference on information reuse and integration (IRI)*, Pittsburgh, PA, USA, 28–30 July 2016, pp. 265–269.
 57. Göngör F and Tutsoy Ö. Design of a chair recognition algorithm and implementation to a humanoid robot. no date.
 58. Matsuno T. Numerical integrations of the primitive equations by a simulated backward difference method. *J Meteorol Soc Jpn Ser II* 1966; 44(1): 76–84. Meteorological Society of Japan.
 59. MATLAB. 9.7.0.1190202 (R2019b). In: *MATLAB:2018*. USA: The MathWorks Inc, 2018.